



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,297	12/31/2003	Richard M. Shupak	MSFT-256&307781.01	1690
41505	7590	10/14/2008		
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)			EXAMINER	
CIRA CENTRE, 12TH FLOOR			SCHMIDT, KARIL	
2929 ARCH STREET			ART UNIT	PAPER NUMBER
PHILADELPHIA, PA 19104-2891			2439	
			MAIL DATE	DELIVERY MODE
			10/14/2008	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/750,297	<b>Applicant(s)</b> SHUPAK ET AL.
	<b>Examiner</b> KARI L. SCHMIDT	<b>Art Unit</b> 2439

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If no period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED. (35 U.S.C. § 133).

Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### **Status**

1) Responsive to communication(s) filed on 27 June 2008.

2a) This action is FINAL.      2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### **Disposition of Claims**

4) Claim(s) 1,2,5,6,10,11,14,15,19,20,23,28,31,33,34 and 36-49 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 1,2,4-6,10,11,14,15,19,20,23,28,31,33,34 and 36 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### **Application Papers**

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on 31 December 2003 is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### **Priority under 35 U.S.C. § 119**

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All    b) Some \* c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### **Attachment(s)**

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-646)

3) Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_

4) Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_

5) Notice of Informal Patent Application

6) Other: \_\_\_\_\_

**DETAILED ACTION**

***Notice to Applicant***

This communication is in response to the amendment filed on 06/27/2008.

Claims 1, 2, 5, 6, 10, 11, 14, 15, 19, 20, 23, 28, 31, 33, 34 and 36-49 remain pending.

Claims 1, 5-6, 10, 14-15, 19, 28, 31, 33-34, and 36 have been amended. Claims 3-4, 7-9, 12-13, 16-18, 21-22, 24-27, 29-30, 32 and 35 have been canceled.

***Response to Arguments***

With regard to Applicant's arguments, it is respectfully submitted that the Examiner has applied new prior art to the amended features of claims (1, 5-6, 10, 14-15, 19, 28, 31, 33-34, and 36). Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

***Claim Objections***

Claims 5-6 are objected to because of the following informalities: The examiner notes claims 5 and 6 are duplicate claims that depend off of claim 1. Appropriate correction is required.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 38-39, 40, 42-43, 44, 46-48, and 49 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

**Claims 38-39, 40, 42-43, 44, 46-48, and 49**

The examiner notes claims 38-39, 40, 42-43, 44, 46-48, and 49 are rejected under 35 U.S.C. 112, second paragraph for being indefinite. Claims 38-39, 40, 42-43, 44, 46-48, and 49 all depend off of a cancelled claim(s) (e.g. 38-39 depend on canceled claim 3, etc). Therefore for the purpose of examination the examiner will interpret these claims to depend off of their respective independent claims (1, 10, and 19).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1, 5-6, 10, 14-15, 19, 28, 31, 33-34, and 36-49 are rejected under 35 U.S.C. 103(a), as best understood, as being unpatentable over Lueh (US 6,658,657 B1) in view of Richarte, Gerardo. "Four different tricks to bypass StackShield and StackGuard protection".

**Claims 1, 10, 19**

Lueh discloses method, system and medium of processing runtime functions, comprising: compiling code to produce executable code (see at least, col. 5, lines 11-14: code is compiled for the first time before execution, which is the native code (e.g. executable code); receiving a call to a runtime function; determining associated data from the call to the runtime function; determining a target address from the associated data; comparing the target address with a reference list of valid target addresses; and if the target address is found on the reference list of valid target addresses then executing the target (see at least, column 2, lines 9-45 : the virtual method gets inlined, the compiler such as JIT compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation.

Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call... and column 4, lines 55-65).

Lueh fails to disclose compiling code to produce executable code that is marked with an identifier indicating that the executable code supports runtime protection; receiving a call to a runtime function of the executable code for a runtime function; and if the target address is not found on the reference list of a valid target addresses then terminating execution of the executable code.

Richarte discloses compiling code to produce executable code that is marked with an identifier indicating that the executable code supports runtime protection (see at least, page 4, 2.2 StackGuard Protection: the examiner notes an output file from the gcc operation is executable code which is marked with a canary (e.g. identifier) for runtime protection); receiving a call to a runtime function of the executable code for a runtime function (see at least, page 4, 2.2. StackGuard Protection: the examiner notes function prologue and epilogue would be contained in a programs body); and if the target address is not found on the reference list of a valid target addresses then terminating execution of the executable code (see at least, 2.3 StackSheild Protection (2.3.2

Checked Clones): the examiner notes the comparison of addresses on a stack and if not matched a SYS\_exit system call is placed (e.g. termination of the executable code)).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh to include compiling code to produce executable code that is marked with an identifier indicating that the executable code supports runtime protection; receiving a call to a runtime function of the executable code for a runtime function; and if the target address is not found on the reference list of a valid target addresses then terminating execution of the executable code as taught by Richarte. One of ordinary skill in the art would have been motivated to combine the teachings in order to provide stack shielding technologies to protect programs against exploitations of stack based buffer overflows (see at least, Richarte, Abstract).

Claims 2, 11, 20

Lueh discloses the method of claim 1, wherein the step of determining the associated data comprises accessing data in a data structure connected with the runtime function and calculating the associated data based on the accessed data (see at least, column 1, lines 12-65, Figures 1 and 2, column 4, lines 45--65).

Claims 5, 6, 14, 15, and 23

Lueh discloses the method of claim 1 comprising the step of generating the reference list of valid target addresses during execution of a previous runtime function (see at least, column 2, lines 46-64: if A's target address of foo is compiled and ran an address

is now known, and therefore the JIT address of A's foo will be used (e.g. occur after a first compilation with A's foo without a known address)).

Claims 28, 31, 34

Lueh discloses the method of claim 1 comprising the step of storing the target address in a caller provided location during execution of a previous runtime function (see at least, column 2, lines 9-45: a vtable is a provided location and if A's target address of foo is compiled and ran an address is now known, and therefore the JIT address of A's foo will be used (e.g. occur after a first compilation with A's foo without a known address)).

Claims 33 and 36

Lueh disclose the method of claim 1 comprising the step of storing the reference list of valid target addresses in memory during the execution of a previous runtime function (see at least, column 2, lines 9-45: a vtable is a provided location and if A's target address of foo is compiled and ran an address is now known, and therefore the JIT address of A's foo will be used (e.g. occur after a first compilation with A's foo without a known address)).

Claims 37, 41, 45

Lueh discloses determining if at least a portion of the associated data is valid (see at least, column 2, lines 9-45 : the virtual method gets inlined, the compiler such as JIT

compiler generates a run-time test to verify if the inlined callee is the right instance to be invoked. The run-time test is typically implemented by checking the vtable or by checking foo of the actual target address of the method invocation. Checking the vtable involves comparing the object's vtable with the vtable of the class of the inlined method. If the comparison is successful (i.e. object matches the vtable of the class of the inlined method) it is safe to execute the inlined code because the inlined method will be dynamically dispatched at runtime. If the comparison fails (i.e. object does not match the vtable of the class of the inlined method) the conventional dispatching code sequence is executed to invoke the virtual method call... and column 4, lines 55-65).

Lueh discloses preventing execution of the target if the associated data is not valid.

Richarte discloses preventing execution of the target if the associated data is not valid (see at least, 3 StackShield Protection (2.3.2 Checked Clones): comparison of addresses on a stack and if not matched a SYS\_exit system call is placed (e.g. termination of the executable code)).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh to include preventing execution of the target if the associated data is not valid as taught by Richarte. One of ordinary skill in the art would have been motivated to combine the teachings in order to provide stack shielding technologies to protect programs against exploitations of stack based buffer overflows (see at least, Richarte, Abstract).

Claims 38, 42, 47

Lueh fails to disclose wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies.

Richarte discloses to wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies (see at least, 2.2. StackGuard Protection: they use pushing and comparing canary (e.g. a secruiyt cookie) on a stack and 2.3 StackShield Protection (2.3.2 Checked Clones): comparison of addresses on a stack and if not matched a SYS\_exit system call is placed (e.g. termination of the executable code)).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Lueh to include wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies as taught by Richarte. One of ordinary skill in the art would have been motivated to combine the teachings in order to provide stack shielding technologies to protect programs against exploitations of stack based buffer overflows (see at least, Richarte, Abstract).

Claims 39, 43, 48

Lueh discloses further comprising determining and storing a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function (see at least, col. 2, lines 46-64: the examiner notes an allocation of memory space is a predetermined calculated value based on the function prior to the runtime call).

Claims 40, 44, 49

Lueh discloses wherein determining if the associated data is valid comprises comparing the predetermined calculated value to another calculated value based on the associated data (see at least, col. 2, lines 9-45: the examiner notes the checking and comparing of the vtable).

Claim 46

Lueh discloses further comprising a storage device that stores a list of valid targets, wherein the dispatcher system determines if the associated data is valid by comparing the target address to the list of valid target addresses (see at least, col. 2, lines 9-45: the examiner notes the checking and comparing of the vtable and col. 4, lines 56-col. 4, line 20: the examiner notes ROM, RAM, disk storage mediums, and etc.).

***Conclusion***

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to KARI L. SCHMIDT whose telephone number is (571) 270-1385. The examiner can normally be reached on Monday - Friday: 7:30am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kambiz Zand can be reached on 571-272-3811. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Kari L Schmidt/  
Examiner, Art Unit 2439

/Kambiz Zand/  
Supervisory Patent Examiner, Art Unit 2434